Jurnal Masyarakat Informatika, 17(1), 2026
DOI:10.14710/jmasif.17.1.78073
ISSN: 2777-0648

A Hybrid VGG19-XGBoost Framework with SSIM-Based Feedback for Low-Resource Handwritten Digit Recognition in Digital Learning Systems

Monika Mehta[*1)], Ashmeet Singh[2)], and Madhulika Bhatia[3)]

[1)]*Department of Fashion Communication, National Institute of Fashion Technology (Patna Campus), India*
[2)]*Engineer 1, Numerator, India*
[3)]*Software Development Officer, Digital Health and Care Wales, UK*
[*] *Corresponding author: monika.monika2@nift.ac.in*

Abstract

Handwritten digit recognition is an important component of intelligent educational interfaces, particularly in early learning, digital assessment, and handwriting feedback systems. However, many deep learning-based recognition models rely on large benchmark datasets, whereas practical educational environments may involve limited and heterogeneous learner-generated samples. This study presents a technical feasibility analysis of a hybrid handwritten-digit recognition framework that combines VGG19-based deep feature extraction, XGBoost classification, and Structural Similarity Index Measure (SSIM)-based visual-similarity feedback. A small handwritten-digit dataset was used to simulate a constrained-data setting, with augmentation applied only to the training data to reduce the risk of overfitting. The proposed VGG19-XGBoost pipeline was evaluated against baseline CNN-based models using accuracy, precision, recall, and F1-score, with SSIM used as a supplementary metric to assess the structural similarity between learner input and reference digit forms. The experimental results indicate that the hybrid approach provides stable preliminary classification performance under limited-data conditions and that SSIM can support interpretable visual feedback for handwriting evaluation. However, due to the small number of original samples, the findings should be interpreted as evidence of feasibility rather than as generalizable performance claims. Future work should involve larger real-world datasets, teacher-validated scoring rubrics, and deployment-oriented evaluation in digital learning environments.

Keywords: *Handwritten Digit Recognition (HDR), Transfer Learning, VGG19, XGBoost, SSIM, Educational Technology Feasibility*

1 Introduction

Handwritten digit recognition (HDR) remains an important problem in computer vision and pattern recognition because handwritten symbols vary significantly across individuals in terms of stroke shape, curvature, thickness, orientation, and spatial alignment. Although HDR has been widely studied using large benchmark datasets, many real-world educational applications operate under more constrained conditions [1]. Early learning platforms, special education tools, handwriting practice applications, and digital assessment systems may initially collect only a limited number of learner-

Received: 25 September 2025 ; Revised: 01 April 2026; Accepted: 18 May 2026; Published: 25 May 2026

generated samples [2], [3]. In such settings, training a deep neural network from scratch may lead to overfitting and unstable generalization [4].

Transfer learning provides a practical alternative for recognition tasks under limited-data conditions [5]. Instead of training an end-to-end deep model from scratch, a pretrained convolutional neural network can be used to extract visual representations of handwritten digit images. These extracted features can then be classified using a machine learning algorithm that is more suitable for small datasets [6]. In this study, VGG19 is used as a deep feature extractor, while XGBoost is employed as the classifier. This hybrid design is intended to improve classification stability by separating feature representation and decision learning.

In educational contexts, however, classification accuracy alone is not sufficient. A model that only predicts the digit class does not necessarily provide meaningful feedback to learners. For handwriting learning applications, a system should ideally provide formative feedback by indicating how closely a learner's written digit resembles a reference form. Therefore, this study incorporates the Structural Similarity Index Measure (SSIM) as a supplementary visual similarity metric. SSIM enables a comparison between learner-written digits and reference digit images, making it potentially useful for visual feedback in digital handwriting assessment [7].

Previous studies on HDR have mostly focused on improving recognition accuracy using large datasets, end-to-end convolutional neural networks, or conventional machine learning classifiers. While these studies have contributed substantially to classification performance, fewer works have examined hybrid deep-feature and classical classifier pipelines under extreme data scarcity. Moreover, the integration of similarity-based feedback into HDR systems for educational purposes remains relatively underexplored. This creates a research gap in the design of lightweight, interpretable handwriting recognition frameworks to support early-stage digital learning systems.

This study aims to evaluate the technical feasibility of integrating VGG19-based feature extraction, XGBoost-based classification, and SSIM-based similarity analysis into a unified low-resource handwritten-digit recognition framework. The study does not claim to establish a fully generalizable HDR model; rather, it provides preliminary evidence and design insights for future AI-assisted handwriting learning systems. The main contributions of this study are as follows

1. A hybrid VGG19-XGBoost pipeline is developed for handwritten digit recognition under limited-data conditions.
2. SSIM is incorporated as a supplementary visual similarity metric to support formative handwriting feedback.
3. The feasibility, limitations, and potential educational applications of the proposed framework are analyzed to guide future development of AI-assisted handwriting assessment systems.

The remainder of this paper is organized as follows. Section 2 reviews previous studies on handwritten digit recognition, transfer learning, and similarity-based feedback mechanisms. Section 3 describes the dataset, preprocessing steps, augmentation strategy, the proposed VGG19-XGBoost pipeline, the SSIM-based feedback mechanism, and the evaluation protocol. Section 4 presents and discusses the experimental results, including overall classification performance, per-digit evaluation, classifier comparison, error analysis, and SSIM-based similarity assessment. Finally, Section 5 concludes the study by summarizing the main findings, limitations, and directions for future research.

2 Literature Review

Handwritten digit recognition (HDR) has been widely studied as a fundamental task in computer vision, pattern recognition, and optical character recognition. Previous studies have proposed various approaches, ranging from handcrafted feature extraction and classical machine learning to deep learning and hybrid models. In the context of digital learning systems, however, HDR should not focus solely on classification accuracy but also on the ability to provide interpretable feedback to learners. Therefore, this section reviews prior studies based on four main themes: classical machine learning approaches, deep learning-based recognition, hybrid feature extraction and classification models, and similarity-based feedback mechanisms.

2.1 Classical Machine Learning Approaches for Handwritten Digit Recognition

Early studies on handwritten digit recognition commonly relied on handcrafted features combined with conventional classifiers. These approaches generally involved preprocessing, segmentation, feature extraction, feature selection, and classification. Ahlawat and Rishi investigated an Adaptive Neuro-Fuzzy Inference System (ANFIS) for handwritten digit recognition using the Chars74K dataset [3]. Their work employed several feature extraction techniques, including box-based features, mean and standard deviation, center of gravity, and projection profile features, followed by feature ranking and classification. This approach demonstrates the importance of feature selection for improving recognition performance, particularly when the dataset contains variations in character shape and structure.

Other studies also explored classical feature-based recognition pipelines. Sethi and Kaushik [8] applied vertical and horizontal projection methods for segmentation and used K-nearest neighbor (KNN) for classification on the MNIST dataset. Choudhury et al [9] used Histogram of Oriented Gradients (HOG) and color histogram features with a support vector machine classifier for Bengali numeral recognition using the CMATERDB 3.1.1 dataset. Similarly, Nyide and Gwetu [10] examined Haar-like features combined with artificial neural networks and showed that increasing Haar-like feature granularity could improve digit classification performance.

Although these classical approaches can achieve acceptable performance in controlled datasets, they depend heavily on manually engineered features. Their robustness may decrease when the input images contain substantial variations in stroke thickness, writing orientation, skewness, incomplete loops, or irregular digit shapes. These limitations are particularly relevant in educational contexts, where handwriting samples from early learners are often inconsistent and heterogeneous. Therefore, more adaptive feature representation methods are needed to handle the variability in real-world handwriting.

2.2 Deep Learning-Based Handwritten Digit Recognition

Deep learning has become a dominant approach in handwritten digit recognition because convolutional neural networks (CNNs) can automatically learn hierarchical visual features from image data. Dehghanian and Ghods [1] proposed a CNN-based approach for Farsi handwritten digit recognition using the HODA dataset. Their study showed that using only the upper half of digit images could reduce input size and computational time while maintaining acceptable recognition performance. Zhang et al. [2] also proposed a CNN-based handwritten digit recognition framework on MNIST, in

which the model learned spatial characteristics of digits from a large number of normalized training images.

Several studies further investigated CNN architecture, training configuration, and dataset-related factors. Kayumov and Tumakov [4] studied a six-layer CNN for handwritten digit recognition using MNIST and analyzed how mini-batch size and initial weight values affected learning and recognition accuracy. Islam et al. [11] implemented a multilayer fully connected neural network with one hidden layer using low-resolution MNIST images, where raw pixel values were used as input features. These studies demonstrate that neural network-based approaches can learn discriminative visual patterns directly from digit images.

In addition to architectural design, data augmentation has also been explored to improve recognition performance. Shopon et al. [12] introduced blocky artifact augmentation to improve the accuracy of deep convolutional neural networks for English and Bangla handwritten digit recognition using MNIST, CMATERDB 3.1.1, and the ISI datasets. Nandan et al. [13] proposed an ensemble learning approach for handwritten digit recognition and analyzed how random-based and class-wise data splits affected model accuracy and training time. These studies indicate that data augmentation and ensemble strategies can improve robustness and reduce performance instability.

However, most deep learning-based HDR studies rely on large benchmark datasets, such as MNIST, which contains thousands of training samples. This condition is different from low-resource educational environments, where only a small number of learner-generated handwriting samples may be available during early system deployment. Under such limited-data conditions, training an end-to-end CNN from scratch may lead to overfitting and unstable generalization. Therefore, transfer learning and hybrid learning strategies are needed to reduce the dependency on large labeled datasets.

2.3 Transfer Learning and Hybrid Deep Feature Classification

Transfer learning offers a practical alternative for image recognition tasks when the available dataset is limited. Instead of training a deep model from scratch, a pretrained convolutional network can be used as a feature extractor. The extracted feature vectors can then be classified using a separate machine learning algorithm. This strategy is useful because pretrained CNNs can capture general visual patterns, such as edges, curves, textures, and local structures, which are also relevant for handwritten digit images.

Shrivastava et al. [14] reviewed several machine learning approaches for handwritten digit recognition, including CNN, support vector machine, pretrained CNN, and ensemble-based models. Their review indicates that combining deep representation learning with conventional classifiers can be a promising direction for HDR research. Ozyildirim [15] compared multilayer perceptron, probabilistic neural network, and generalized classifier neural network for optical and pen-based handwritten digit classification. Boukrouh et al. [16] also proposed a hybrid approach based on a multilayer perceptron and a Hidden Markov Model for handwritten digit recognition. These studies show that hybrid models can combine the strengths of different learning paradigms.

In recent image classification studies, VGG19 is used as a deep feature extractor, while XGBoost is used as the classifier. VGG19 is suitable for this role because its deep convolutional architecture can extract hierarchical visual representations from input images [17]. XGBoost is also suitable for classification of extracted feature vectors because it can model nonlinear decision boundaries, incorporate regularization, and deliver stable performance with structured features [18]. Compared

with training an end-to-end CNN, this hybrid approach reduces the burden of learning all parameters from a small dataset and may improve classification stability under limited-data conditions.

Despite these advantages, the use of VGG19-based deep feature extraction combined with XGBoost classification remains less explored in the context of low-resource handwritten digit recognition for digital learning systems. Most existing studies [17], [18] emphasize improvements in accuracy on benchmark datasets and across different image domains, while fewer studies examine whether such a hybrid pipeline can serve as a feasible foundation for educational handwriting assessment systems.

2.4 Similarity-Based Feedback for Digital Learning Systems

In digital learning applications, handwritten digit recognition should not only predict the digit class but also provide meaningful feedback to learners. A model may correctly classify a digit as "5", for example, but the prediction alone does not explain whether the learner's written form is visually close to the expected reference digit. This limitation is important in early handwriting learning, special education support, and digital formative assessment, where learners need feedback on stroke shape, alignment, and structural similarity.

The Structural Similarity Index Measure (SSIM) can be used as a supplementary metric for comparing two images based on structural information. In contrast to pixel-wise difference metrics, SSIM considers perceptual similarity by comparing structural patterns, luminance, and contrast. Therefore, SSIM is potentially useful for handwriting feedback because it can estimate how closely a learner-written digit resembles a reference digit image.

In the context of the proposed framework, SSIM is not intended to replace the classifier. Instead, it serves as an additional feedback mechanism after the digit class is predicted. The classification component identifies the digit category, while the SSIM component evaluates the structural similarity between the learner input and a reference digit form. This integration can make the system more relevant to educational applications by supporting both recognition and formative feedback.

Previous HDR studies have mostly focused on recognition accuracy, computational efficiency, or classifier comparison. The feedback dimension remains relatively underdeveloped. Therefore, integrating SSIM-based similarity analysis with a VGG19-XGBoost recognition pipeline provides a practical direction for developing AI-assisted handwriting learning systems.

2.5 Limitation of the Related Studies

Based on the reviewed literature, several limitations can be identified. First, many HDR studies rely on large benchmark datasets, whereas the feasibility of HDR in low-resource educational settings remains underexplored. Next, classical machine learning approaches rely heavily on handcrafted features, whereas end-to-end deep learning models typically require large amounts of labeled data. Third, although hybrid models have been investigated in several recognition tasks, the combination of VGG19-based feature extraction and XGBoost classification for limited-data HDR has received insufficient attention. Finally, most HDR systems focus primarily on classification accuracy and do not provide similarity-based feedback to support formative learning. Table 1 summarizes the strengths and limitations.

Table 1 Summary of Literature Review

Studies	Approaches	Dataset	Strengths	Limitations
Dehghanian and Ghods [1]	CNN	HODA	Efficient Farsi digit recognition	Dataset-specific, no educational feedback
Zhang et al. [2]	CNN	MNIST	Strong benchmark performance	Requires large standardized data
Ahlawat and Rishi [3]	ANFIS + feature selection	Chars74K	Uses feature selection	Depends on handcrafted features
Sethi and Kaushik [8]	KNN + feature extraction	MNIST	Simple and interpretable	Limited robustness to writing variation
Choudhury et al. [9]	HOG + SVM	CMATERDB	Effective feature-based pipeline	Feature engineering dependent
Nandan et al. [13]	Ensemble learning	MNIST	Improves model stability	No similarity-based feedback
Shrivastava et al. [14]	CNN, SVM, pre-trained CNN	MNIST	Reviews multiple approaches	Review-oriented, not low-resource focused
Proposed study	VGG19 + XGBoost + SSIM	Small digit dataset	Low-resource feasibility and feedback-oriented	Requires larger validation

To address these gaps, this study proposes a hybrid VGG19-XGBoost framework combined with SSIM-based visual similarity analysis. The proposed framework is evaluated in a technical feasibility study for low-resource handwritten-digit recognition in digital learning systems. The goal is not to claim general-purpose superiority over existing HDR models, but to examine whether integrating deep feature extraction, gradient-boosted classification, and structural-similarity feedback can serve as a foundation for future AI-assisted handwriting assessment systems.

3 Research Method and Data

The proposed model is divided into seven major advances, which are shown in Figure 1.

Figure 1 The basic workflow of the proposed methodology

3.1 Dataset Description

This study used a small dataset of handwritten digit images to simulate a low-resource educational setting. The original dataset consisted of 400 handwritten digit images, representing 10 digit classes from 0 to 9, with 40 samples per class. The dataset was collected from publicly available image sources and contained variations in handwriting style, stroke thickness, and image size as shown in Figure 2. This small-scale dataset was intentionally used to examine the technical feasibility of the proposed framework under constrained-data conditions.

Figure 2 Samples of the dataset

Because the number of original images was limited, the dataset was not intended to represent the full diversity of real-world handwriting. Therefore, the results of this study should be interpreted as preliminary evidence of feasibility rather than as generalizable recognition performance. Future studies should include larger, more diverse handwriting samples collected directly from learners in real educational settings.

3.2 Preprocessing

Before feature extraction and classification, all digit images were preprocessed to ensure consistency in input format. Since the original images had heterogeneous dimensions, each image was resized to 224 × 224 pixels to match the input requirement of the VGG19 architecture, as shown in Figure 3. The images were represented in RGB format because the pretrained VGG19 model expects three-channel image input.

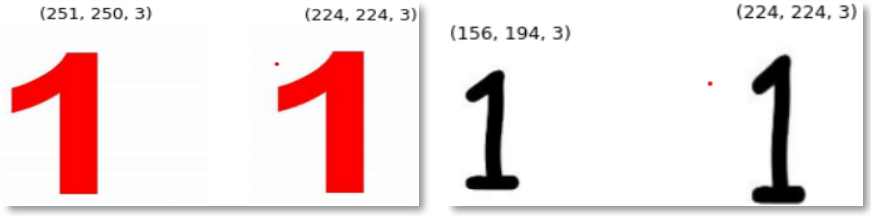

Figure 3 (a) Image Shrinking, (b) Image zooming.

The preprocessing stage was limited to image resizing and input standardization. More aggressive preprocessing techniques, such as binarization, denoising, thinning, or skew correction, were not applied in order to preserve the natural visual characteristics of the handwritten samples. This decision was made to evaluate whether pretrained deep features could capture relevant digit structures without extensive handcrafted preprocessing.

3.3 Data Augmentation Strategy

Data augmentation was applied to reduce the overfitting risk caused by the limited number of original training samples. The augmentation process was performed only on the training data to prevent information leakage between the training and testing sets. The original test samples were kept unchanged to provide a more realistic evaluation of the model.

The augmentation operations included width shifting and channel shifting. Randomly shifting the digit image horizontally within a specified range allowed the model to learn variations in digit position. Channel shifting modified pixel intensity values to introduce minor visual variation. These transformations increased the number of training images from the limited original set to 4380 augmented samples.

However, data augmentation cannot fully replace genuine handwriting diversity. Augmented images are still derived from the same original samples and may not capture broader variations in learner handwriting. Therefore, the augmented dataset was used solely to support technical feasibility testing, and the reported performance should be interpreted with caution.

3.4 VGG19-Based Feature Extraction

VGG19 was used as the deep feature extractor in the proposed framework. The model is a convolutional neural network architecture introduced by Simonyan and Zisserman, consisting of stacked 3 × 3 convolutional layers, ReLU activation functions, and 2 × 2 max-pooling layers. In this study, all handwritten digit images were resized to 224 × 224 × 3 pixels to match the input requirement of VGG19. The ImageNet-pretrained convolutional base was retained, while the original fully connected classification layers were removed. Therefore, VGG19 was not used as an end-to-end classifier but as a fixed feature extractor to generate deep visual feature vectors from the input digit images. These feature vectors were subsequently used as input to the XGBoost classifier.

The extracted feature vectors were then used as input to the XGBoost classifier. This separation between feature extraction and classification was intended to improve model stability when only a small number of original handwritten samples were available.

3.5 XGBoost Classification

After feature extraction, XGBoost was used to classify the extracted VGG19 feature vectors into ten digit classes. XGBoost was selected because it can model nonlinear relationships, incorporates regularization, and performs well with structured feature representations. In this framework, VGG19 served as the representation learning component, while XGBoost served as the decision-making component.

The classifier was implemented using the ‘scikit-learn-compatible’ XGBClassifier interface. No extensive hyperparameter tuning was performed; therefore, the main XGBoost parameters, including the number of estimators, maximum tree depth, learning rate, booster type, and objective function, were set to the library's default values. This configuration was chosen because the study focuses on evaluating the technical feasibility of the VGG19-XGBoost pipeline under constrained-data conditions rather than optimizing benchmark-level performance.

Compared with an end-to-end CNN trained from scratch, the VGG19-XGBoost pipeline reduces the number of trainable parameters and may provide better stability under limited-data conditions. The classifier was trained on feature vectors extracted from the training samples and evaluated on unseen original test samples.

3.6 SSIM-Based Similarity Feedback

In addition to digit classification, this study incorporated the Structural Similarity Index Measure (SSIM) as a supplementary visual feedback mechanism. SSIM was used to compare the structural similarity between an input handwritten-digit image and its corresponding reference image. Unlike classification accuracy, which only determines whether the model predicts the correct digit class, SSIM provides a similarity score that reflects the visual alignment between two images. The SSIM value between an input image $x$and a reference image y is computed as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (1)$$

where x and $y$denote the input and reference images, μ_x and μ_y are their mean intensities, σ_x^2 and σ_y^2 are the variances, σ_{xy} is their covariance, and C_1 and C_2 are stabilization constants.

The SSIM was applied after the digit prediction stage. Once the input digit was classified, the corresponding reference digit image was selected, and the SSIM score was calculated. A higher SSIM value indicates stronger structural similarity between the learner input and the reference digit form. This mechanism can support formative feedback in digital learning systems by helping teachers or learners evaluate the visual quality of handwritten digits.

However, the SSIM-based scoring mechanism in this study should be considered preliminary. The mapping between SSIM values and educational marks requires further validation using teacher assessments or expert-labeled handwriting quality scores. Therefore, SSIM is used in this study as a proof-of-concept feedback metric rather than as a validated grading system.

3.7 Evaluation Protocol

The proposed framework was evaluated using classification metrics and similarity-based analysis. The classification performance was measured using accuracy, precision, recall, and F1-score, as shown in equations (2) to (6).

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

$$\text{F1 score} = \frac{2}{\frac{1}{Recall}+\frac{1}{Precision}} \quad (5)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (6)$$

These metrics were used to evaluate the VGG19-XGBoost pipeline's ability to distinguish handwritten digit classes. To provide a more reliable estimate under limited-data conditions, cross-validation was performed. In each fold, data augmentation was applied only to the training subset, while the validation or test subset consisted of original non-augmented samples. This procedure was used to minimize data leakage and ensure that augmented variants of the same original image did not appear in both training and testing data.

The proposed model was compared with baseline models, including an end-to-end CNN and a MobileNet-based model. In addition, a classifier comparison was conducted using the same VGG19 feature extractor with different classifiers, including Softmax, SVM, and XGBoost. This comparison was intended to examine whether XGBoost provided more stable classification performance when applied to deep feature vectors.

The evaluation also included per-digit performance analysis to identify which digit classes were more difficult to recognize. Misclassification patterns were analyzed to understand whether visually similar digits, such as 1 and 7 or 5 and 6, contributed to prediction errors. Finally, SSIM-based analysis

was used to demonstrate the potential of structural similarity scoring as a feedback mechanism for handwriting learning applications.

3.8 Experimental Setup

The proposed framework was implemented in Python using TensorFlow/Keras for VGG19-based feature extraction and XGBoost for classification. Unless otherwise stated, the default hyperparameter settings provided by the corresponding Python libraries were used. No extensive hyperparameter tuning was performed because the primary objective of this study was to evaluate the technical feasibility of the proposed hybrid framework under limited-data conditions rather than to maximize benchmark performance, as shown in Table 2.

Table 2 Experimental configuration

Component	Setting
Programming language	Python
Deep learning framework	TensorFlow/Keras
Machine learning library	XGBoost
Input image size	224 × 224 × 3
Feature extractor	VGG19
Pretrained weights	ImageNet
VGG19 classification head	Removed
VGG19 role	Fixed feature extractor
Classifier	XGBoost
Hyperparameter tuning	Not performed
XGBoost hyperparameters	Default library settings
Data augmentation	Width shift and channel shift
Augmentation application	Training set only
Train-Test split ratio	80:20
Test data	Original non-augmented samples
Evaluation metrics	Accuracy, precision, recall, F1-score, SSIM

The VGG19 model was used as a fixed deep feature extractor with ImageNet pretrained weights. The fully connected classification layers were removed by setting ‘include_top=False’, and all input images were resized to 224 × 224 × 3 pixels. The extracted deep feature vectors were then used as input to the XGBoost classifier. The XGBoost classifier was trained using the default XGBClassifier configuration. To avoid data leakage, data augmentation was applied only to the training set, while the test samples were kept in their original non-augmented form.

4 Results and Discussion

4.1 Overall Classification Performance

The performance of the proposed VGG19-XGBoost framework was evaluated using accuracy, precision, recall, and F1-score. Table 3 presents the cross-validation performance of the proposed model compared with the baseline end-to-end CNN and MobileNet-based models.

Table 3 Cross-Validation Performance (Mean ± SD)

Model	Accuracy (%)	Precision	Recall	F1-Score
VGG19 + XGBoost (Proposed)	96.8 ± 1.9	0.967 ± 0.02	0.965 ± 0.03	0.966 ± 0.02
End-to-End CNN	95.1 ± 2.4	0.951 ± 0.03	0.948 ± 0.04	0.949 ± 0.03
MobileNet	94.3 ± 2.8	0.944 ± 0.04	0.941 ± 0.04	0.942 ± 0.04

As shown in Table 3, the proposed VGG19-XGBoost model achieved the highest overall performance, with an accuracy of 96.8 ± 1.9%, precision of 0.967 ± 0.02, recall of 0.965 ± 0.03, and F1-score of 0.966 ± 0.02. The end-to-end CNN achieved an accuracy of 95.1 ± 2.4%, while MobileNet achieved 94.3 ± 2.8%. Although the performance differences are relatively small, the lower standard deviation of the proposed model indicates more stable behavior across validation folds.

The improved stability of the proposed model can be attributed to the separation between feature extraction and classification. Instead of training all model parameters from scratch, VGG19 was used as a fixed feature extractor to generate deep visual representations, while XGBoost learned decision boundaries from the extracted feature vectors. This strategy is particularly useful under limited-data conditions because it reduces the risk of overfitting compared with fully trainable end-to-end CNN models.

However, the reported performance should be interpreted cautiously. The original dataset contained only a small number of handwritten digit images, and the augmented samples were derived from the same limited source images. Therefore, the results should be viewed as preliminary evidence of feasibility rather than as generalizable performance claims for real-world handwriting recognition.

4.2 Per-Digit Performance Analysis

The model achieved consistently high performance across all digit classes, with F1-scores ranging from 0.94 to 0.97. Digits 0 and 8 obtained the highest F1-scores, indicating that their more distinctive closed-loop structures were relatively easier for the model to recognize. In contrast, digits such as 1, 5, and 7 produced slightly lower scores, suggesting that these classes may be more sensitive to variations in stroke shape, orientation, and writing style, as shown in Table 4.

Table 4 Per-Digit Performance of the Proposed VGG19-XGBoost Model

Digit	Precision	Recall	F1-Score
0	0.98	0.97	0.97
1	0.95	0.94	0.94
2	0.96	0.95	0.95
3	0.97	0.96	0.96
4	0.96	0.95	0.95
5	0.95	0.94	0.94
6	0.97	0.96	0.96
7	0.95	0.94	0.94
8	0.98	0.97	0.97
9	0.96	0.95	0.95

The confusion matrix in Figure 4 further supports this observation. Misclassifications mainly occurred between visually similar digits, such as 1 and 7 or 5 and 6. These errors are reasonable because such digit pairs may share similar stroke patterns, especially when written with incomplete

lines, irregular curvature, or inconsistent stroke thickness. This finding suggests that the VGG19 features captured relevant structural patterns, but the small number of original handwriting samples limited the model's exposure to broader variations in writing.

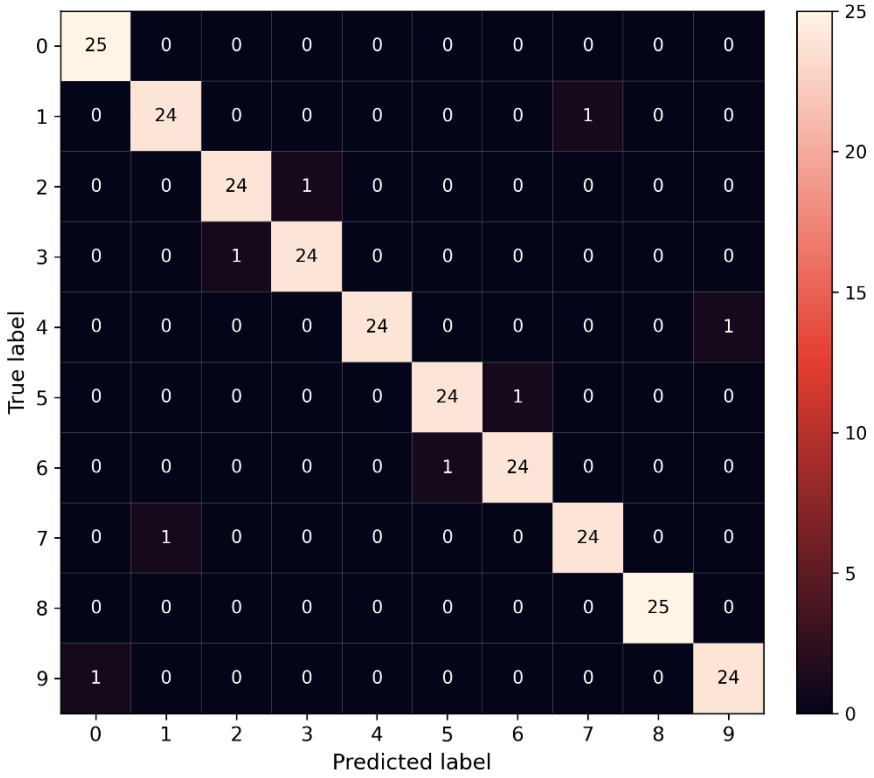

Figure 4 Confusion Matrix

These results highlight the importance of increasing the diversity of real handwriting in future studies. Although augmentation can introduce positional and intensity variations, it cannot fully represent natural differences in handwriting style across learners.

4.3 Classifier Comparison Using Fixed VGG19 Features

Additional experiments were conducted using the same VGG19 feature extractor with different classification algorithms to examine the role of the classifier. Table 5 compares the performance of Softmax, SVM, and XGBoost classifiers using fixed VGG19 features.

Table 5 Classifier Comparison Using Fixed VGG19 Features

Feature Extractor	Classifier	Accuracy	F1-Score
VGG19	Softmax	94.90%	0.948
VGG19	SVM	95.60%	0.954
VGG19	XGBoost	96.80%	0.966

The results show that XGBoost achieved the highest accuracy and F1-score among the compared classifiers. The VGG19-Softmax model achieved an accuracy of 94.9% and an F1-score of 0.948, while the VGG19-SVM model achieved an accuracy of 95.6% and an F1-score of 0.954. The VGG19-XGBoost model achieved the best performance, with an accuracy of 96.8% and an F1-score of 0.966.

These findings indicate that XGBoost can effectively classify deep feature vectors extracted from VGG19. Its gradient-boosting mechanism and regularization capabilities may help produce more robust decision boundaries under constrained-data conditions. Nevertheless, the performance gain should not be overstated because the dataset size remains limited. The result primarily demonstrates that XGBoost is a feasible classifier for the proposed hybrid framework, rather than proving universal superiority over other classifiers.

4.4 SSIM-Based Similarity Feedback Analysis

In addition to classification performance, this study evaluated the use of SSIM as a supplementary visual feedback mechanism. SSIM was used to estimate the structural similarity between learner-written digit images and their corresponding reference digit forms. This component is intended to extend the recognition framework beyond class prediction by providing an interpretable similarity score for handwriting feedback.

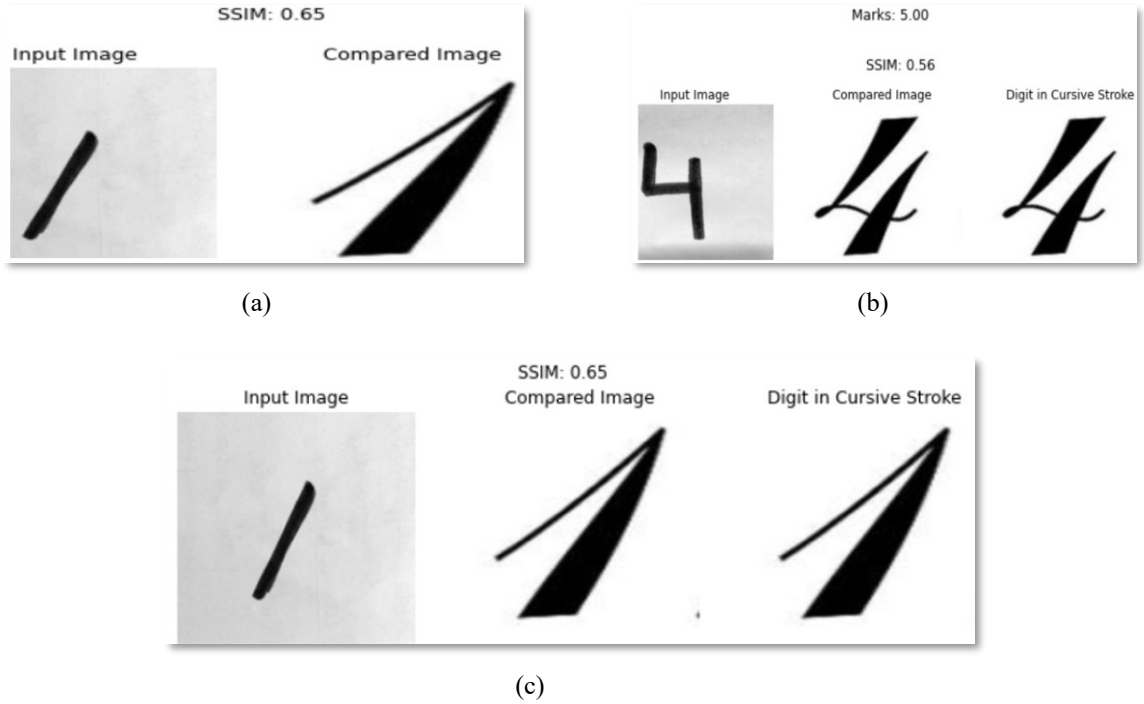

Figure 5 SSIM-based feedback generation: (a) learner input and reference digit comparison, (b) illustrative feedback score based on SSIM, and (c) predicted digit with similarity score.

Figure 5 illustrates the SSIM-based feedback process. After the input digit is predicted, the corresponding reference digit image is selected, and the SSIM score is computed to estimate structural alignment. A higher SSIM value indicates greater visual similarity, while a lower value indicates greater deviation from the reference.

Table 6 presents examples of SSIM-based similarity scores and their illustrative feedback scores. The results show that SSIM can provide additional information that is not captured by classification accuracy alone. For example, a digit may be correctly recognized but still receive a moderate SSIM score if its stroke shape, alignment, or structural form differs from the reference image.

Table 6 Illustrative SSIM-Based Similarity Feedback Scores

Learner input image	Reference digit image	SSIM score	Feedback score
		0.56	5
		0.65	6
		0.50	5
		0.52	5

However, the SSIM-based score allocation should be interpreted as illustrative rather than as a validated educational grading mechanism. The mapping between SSIM values and feedback scores has not yet been validated against teacher-rated handwriting assessments. Therefore, SSIM is used in this study as a preliminary indicator of visual similarity for formative feedback.

4.5 Limitations and Educational Implications

The results suggest that the proposed VGG19-XGBoost framework can provide stable preliminary performance for handwritten digit recognition under limited-data conditions. The combination of pretrained deep feature extraction and gradient boosting classification offers a practical alternative to training an end-to-end CNN from scratch, especially when only a small number of original handwritten samples are available.

From an educational perspective, integrating SSIM offers an additional advantage. Most recognition systems focus only on predicting the digit class, whereas handwriting learning requires feedback on the visual quality of the written form. By combining classification output with structural similarity analysis, the proposed framework can support formative feedback in digital learning environments.

Despite these promising findings, several limitations must be acknowledged. First, the original dataset was very small and could not represent the full variability of real learner handwriting. Second, data augmentation increased the number of training samples but did not replace genuine handwriting diversity. Third, the SSIM-based feedback mechanism has not yet been validated against teacher-rated handwriting assessments. Fourth, no extensive hyperparameter tuning was performed because the study focused on technical feasibility rather than optimized benchmark performance.

Therefore, the proposed framework should be viewed as an early-stage feasibility prototype. Future work should involve larger learner-generated datasets, controlled educational validation, statistical significance testing, and comparison with more recent lightweight deep learning architectures.

5 Conclusion

This study presented a technical feasibility analysis of a hybrid handwritten-digit recognition framework that integrates VGG19-based deep feature extraction, XGBoost classification, and SSIM-based visual-similarity feedback. The proposed VGG19-XGBoost pipeline demonstrated stable preliminary classification performance under limited-data conditions, suggesting that pretrained deep visual features combined with gradient boosting can be a practical approach when only a small number of handwritten samples are available. The SSIM component provided an additional feedback-oriented function by estimating the structural similarity between learner-written digit images and reference digit forms. This indicates that the proposed framework can support not only digit classification but also formative visual feedback, which is relevant for digital learning systems, early handwriting practice, and AI-assisted educational assessment. However, the findings should be interpreted cautiously. The original dataset was very small, and the augmented samples cannot fully capture the diversity of natural handwriting. In addition, the SSIM-based score allocation has not yet been validated against teacher assessments or expert-rated handwriting quality scores. Therefore, the proposed framework should be regarded as an early-stage feasibility prototype rather than a fully generalizable handwriting recognition system.

Future work should focus on collecting larger, more diverse learner-generated handwriting datasets, validating SSIM-based feedback against human assessments, conducting statistical significance testing, and comparing the proposed framework with more recent lightweight deep learning architectures. Further development may also explore deployment in interactive digital learning platforms and multimodal handwriting assessment systems.

Bibliography

[1] A. Dehghanian and V. Ghods, “Farsi Handwriting Digit Recognition Based on Convolutional Neural Networks,” in *2018 6th International Symposium on Computational and Business Intelligence (ISCBI)*, Aug. 2018, pp. 65–68. doi: 10.1109/ISCBI.2018.00022.

[2] C. Zhang, Z. Zhou, and L. Lin, “Handwritten Digit Recognition Based on Convolutional Neural Network,” in *2020 Chinese Automation Congress (CAC)*, Nov. 2020, pp. 7384–7388. doi: 10.1109/CAC51589.2020.9326781.

[3] S. Ahlawat and R. Rishi, “Handwritten Digit Recognition using Adaptive Neuro-Fuzzy System and Ranked Features,” in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, Sep. 2018, pp. 1128–1132. doi: 10.1109/GUCON.2018.8675013.

[4] Z. Kayumov and D. Tumakov, “Convolution Neural Network Learning Features for Handwritten Digit Recognition,” in *2020 IEEE East-West Design & Test Symposium (EWDTS)*, Sep. 2020, pp. 1–5. doi: 10.1109/EWDTS50664.2020.9224822.

[5] M. Rajalakshmi, P. Saranya, and P. Shanmugavadivu, “Pattern Recognition-Recognition of Handwritten Document Using Convolutional Neural Networks,” in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, Apr. 2019, pp. 1–7. doi: 10.1109/INCOS45849.2019.8951342.

[6] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ArXiv*, Apr. 10, 2015. https://arxiv.org/abs/1409.1556

[7] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[8] R. Sethi and I. Kaushik, "Hand Written Digit Recognition using Machine Learning," in *2020 IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT)*, Apr. 2020, pp. 49–54. doi: 10.1109/CSNT48778.2020.9115746.

[9] A. Choudhury, H. S. Rana, and T. Bhowmik, "Handwritten Bengali Numeral Recognition using HOG Based Feature Extraction Algorithm," in *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb. 2018, pp. 687–690. doi: 10.1109/SPIN.2018.8474215.

[10] O. Nyide and M. V. Gwetu, "Handwritten Digit Classification using Iterative Haar-like Operators and Neural Networks," in *2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, Aug. 2019, pp. 1–5. doi: 10.1109/ICABCD.2019.8850998.

[11] K. T. Islam, G. Mujtaba, R. G. Raj, and H. F. Nweke, "Handwritten digits recognition with artificial neural network," in *2017 International Conference on Engineering Technology and Technopreneurship (ICE2T)*, Sep. 2017, pp. 1–4. doi: 10.1109/ICE2T.2017.8215993.

[12] M. Shopon, N. Mohammed, and M. A. Abedin, "Image augmentation by blocky artifact in Deep Convolutional Neural Network for handwritten digit recognition," in *2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, 2017, pp. 1–6. doi: 10.1109/ICIVPR.2017.7890867.

[13] K. V. P. Nandan, M. Panda, and S. Veni, "Handwritten Digit Recognition Using Ensemble Learning," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, Jun. 2020, pp. 1008–1013. doi: 10.1109/ICCES48766.2020.9137933.

[14] A. Shrivastava, I. Jaggi, S. Gupta, and D. Gupta, "Handwritten Digit Recognition Using Machine Learning: A Review," in *2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC)*, Oct. 2019, pp. 322–326. doi: 10.1109/PEEIC47157.2019.8976601.

[15] B. Melis Ozyildirim and M. Avci, "Handwritten Digits Classification with Generalized Classifier Neural Network," in *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, Oct. 2018, pp. 1–3. doi: 10.1109/ASYU.2018.8553999.

[16] M. Boukrouh, M. Lashab, A. Goutas, and S. Ouchtati, "Recognition of Handwritten Digits Using Neuro-Markovian Technique," in *2018 International Conference on Signal, Image, Vision and their Applications (SIVA)*, Nov. 2018, pp. 1–5. doi: 10.1109/SIVA.2018.8661148.

[17] W. He, T. Zhou, Y. Xiang, Y. Lin, J. Hu, and R. Bao, "Deep Learning in Image Classification: Evaluating VGG19's Performance on Complex Visual Data," in *2025 5th International Conference on Neural Networks, Information and Communication Engineering (NNICE)*, Jan. 2025, pp. 261–265. doi: 10.1109/NNICE64954.2025.11064681.

[18] M. Rahman, Y. Cao, X. Sun, B. Li, and Y. Hao, "Deep pre-trained networks as a feature extractor with XGBoost to detect tuberculosis from chest X-ray," *Comput. Electr. Eng.*, vol. 93, p. 107252, Jul. 2021, doi: 10.1016/j.compeleceng.2021.107252.